REMARKS

In the aforementioned Office Action, claims 1-26 were examined and rejected. In view of the foregoing amendments and the following remarks, Applicants hereby respectfully request reconsideration of the Application.

An appendix with a marked-up version of the above amendments follows at the end of this response.

Rejection under 35 U.S.C. § 102

In paragraph 3 of the Office Action, the Examiner rejected claims 1-26 under 35 U.S.C. § 102(e) as being anticipated by Larson (U.S. Patent No. 6,359,623). Applicant respectfully traverses.

Claim 1 recites, in part, "a graphics engine coupled to the memory and including a **pipeline structure** ..." (emphasis added). In support of the invention as claimed, the specification states, paragraph 68, lines 1-3, "graphics engine 1006 is a pipeline structure designed to carry out a subdivision process both sequentially and in **parallel**" (emphasis added). Additionally, the specification states, paragraph 72, lines 1-3, "[b]y implementing a pipeline structure ... in which some subdivisions are carried out in parallel, a number of cycles required to determine whether pixels in a tile are inside or outside a primitive is greatly reduced."

Applicant respectfully submits that Larson does not disclose, teach, or suggest a graphics engine including a pipeline structure, based at least upon the fact that Larson does not disclose a system or method in which subdivisions are carried out in parallel, or in which any processes are carried out in parallel. Applicant submits that although Larson states, col. 6, lines 19-20, "polygons are being processed through the graphics pipeline," Larson's "graphics pipeline" is not a "pipeline structure," as claimed in the present

invention. A "pipeline structure" of the present invention refers to a particular architectural design of the graphics engine to accommodate parallel processing. In contrast, Larson uses the phrase "graphics pipeline" to signify that polygons are being processed in a sequential manner. In other words, Larson does not disclose a "pipeline structure." In fact, Larson teaches away from a pipeline structure and parallel architecture. Larson states, col. 10, lines 46-49, "logic can be configured in the hierarchical tiler 57 to implement **nested loops** for the x and y screen space coordinates to group the pixels of the region ..." (emphasis added). Applicant submits that implementation of nested loops are not conducive to parallel processing in a parallel architecture.

Furthermore, claim 1 recites, in part, "a memory for storing a raster image ... the pipeline structure receiving information related to polygonal portions of the raster image from the memory ...." Larson discloses that RAM 45 (FIG. 2) stores pixel colors and Z coordinates (col. 5, lines 63-67). However, Applicant submits that Larson does not disclose "a memory for storing a **raster image**," a memory for storing "information related to polygonal portions of the raster image," or a "pipeline structure receiving information ... from the memory," as claimed in claim 1.

Based at least upon the above remarks, Applicant submits that independent claim 1 is not anticipated by Larson, and requests that claim 1 be allowed.

Regarding claim 2, the Examiner states that Larson discloses, col. 3, lines 17-37, a predetermined number of parallel logic circuits. Applicant submits that Larson discloses a hierarchical tiler that (1) subdivides a screen space into a plurality of regions, (2) determines whether each of the regions are entirely inside, entirely outside, or partially inside of a primitive, and (3) converts those regions that are entirely inside the primitive into pixel locations in screen space. The hierarchical tiler then repeats the above steps for

each region that is determined to be partially inside the primitive. Applicant respectfully submits that Larson does not teach or disclose a predetermined number of parallel logic circuits, as claimed. That is, Larson does not disclose a hierarchical tiler that processes information in parallel, or that comprises parallel logic circuits. Therefore, based at least upon the above remarks, Applicant submits that claim 2 is not anticipated by Larson, and respectfully requests that claim 2 be allowed. Furthermore, since claims 3-11 depend directly or indirectly from claim 1, Applicant submits that claims 3-11 are allowable for at least the same reasons as claim 1.

Claim 14 recites, in part, "having a coordinate reference frame located at a geometric center of the polygonal portion ...." Applicant respectfully submits that the Examiner has not shown, nor has Larson disclosed, a method of determining whether a polygonal portion of a raster image is at least partly inside a graphics primitive, in which a coordinate reference frame is located at a geometric center of the polygonal portion, as claimed. In fact, Larson discloses a method in which the coordinate reference frame is **not** located at the geometric center of a polygonal portion, but instead at a corner vertex (X_0, Y_0) (FIG. 11) of a primitive 212 (FIG. 11). In contrast, the specification of the present invention discloses a reference frame located at the geometric center of a tile R (i.e., polygonal portion), as illustrated in FIG. 8. Based at least upon the above reasons, Applicant respectfully submits that claim 14, dependent from claim 12, is not anticipated by Larson. Therefore, Applicant is amending claim 12 to include the limitation of claim 14 as discussed above. Applicant now believes that claim 12, as amended, is allowable. In addition, Applicant is amending claim 14 accordingly. Since claims 13, 15-19 and amended claim 14 depend directly or indirectly from claim 12, Applicant respectfully

requests that claims 13, 15-19 and amended claim 14 be allowed for at least the same reasons as amended claim 12.

Applicant is amending independent claim 20 to be similar in scope to amended claim 12, and amending independent claims 22 and 26 by including the limitation "defining a coordinate reference frame located at a geometric center of the tile." Applicant submits that amended claims 20, 22 and 26 recite no new material, and respectfully requests that claims 20, 22 and 26 be allowed, based on at least the reasons above in conjunction with amended claim 12. Furthermore, since claim 21 depends from amended claim 20, and claims 23-25 depend from amended claim 22, Applicant requests that claims 21 and 23-25 be allowed for at least the same reasons as claims 20 and 22, respectively.

Based on the foregoing remarks, Applicant believes that the rejections in the Office Action of April 9, 2003 are fully overcome, and that the Application is in condition for allowance. If the Examiner has questions regarding the case, he is invited to contact Applicant's undersigned representative at the number given below.

Respectfully submitted,

Daniel H. McCabe

Date: July 9, 2003 By: Susan Yee

Susan Yee, Reg. No. 41,388
Carr & Ferrell LLP
2225 East Bayshore Road, Suite 200
Palo Alto, CA 94303
Phone: (650) 812-3400
Fax: (650) 812-3444

APPENDIX

In the specification:

Paragraph 1:

This application claims the benefit of Provisional Patent Application Serial No. 60/204,203, filed on May 15, 2000, entitled "Parallel Architecture for Graphics Primitives Decomposition," which is incorporated herein by reference. The present application also is related to and incorporates by reference co-pending U.S. Patent Application Serial No. [_____] 09/858,306, by Daniel H. McCabe, entitled "Graphics Primitive Decomposition Using Edge Functions and Recursive Tile Subdivision", filed on May 15, 2001.

Paragraph 54:

To determine the tile corner farthest in the positive direction from the edge without having to perform the edge function, steps are taken to advantageously use a tile's symmetry. By placing a reference coordinate system at a center of the tile, the general edge function $e(x, y) = e_0 + n_x x + n_y y$ (equation (9)), provides a largest positive value when the terms $n_x x$ and $n_y y$ are absolute numbers given that e_0 is a constant. Thus in FIG. 8, a reference coordinate system 800 is placed in a center of tile R. Subsequently, coordinates of the four corners of tile R are ($-\Delta X/2, -\Delta Y/[1]2$), ($\Delta X/2, -\Delta Y/2$), ($-\Delta X/2, \Delta Y/2$) and ($\Delta X/2, \Delta Y/2$). As such, regardless of whether actual values of n_x and n_y are positive or negative, there exists a tile corner having a coordinate combination such that when the terms $n_x x$ and $n_y y$ are computed, results are positive. Given that the term e_0 is a constant, having positive terms $n_x x$ and $n_y y$ results in the largest value for an edge function.

Accordingly, the edge function at the tile corner that is farthest in the positive direction from an edge can be evaluated by performing only one calculation.

Paragraph 73:

The sequential logic circuit 1102 receives as input primitive edge functions and tile corner coordinates from the CPU 402 and memory 1014, respectively. The sequential logic circuit 1102 evaluates the edge functions at each corner of a tile to determine whether the tile is inside of a primitive. This evaluation is carried out according to the discussion corresponding to FIG. 8. If a tile is outside the primitive, the tile is disregarded from subsequent decomposition, and the graphics processor 100[0]6 (via sequential logic circuit 1102), subsequently, receives new corner coordinates for a new tile.

In the claims:

12. (Once Amended) A method of identifying pixels inside a graphics primitive of a raster image, comprising the steps of:

(a) determining whether a polygonal portion of the raster image is at least partly inside the graphics primitive by using a coordinate reference frame located at a geometric center of the polygonal portion;

(b) dividing the polygonal portion of the raster image into a predetermined number of polygonal subportions if the polygonal portion of the raster image is at least partly inside the graphics primitive;

(c) determining whether each polygonal subportion of the raster image is at least partly inside the graphics primitive; and

(d) further dividing the polygonal subportion into a predetermined number of polygonal subportions if the polygonal subportion is at least partly inside the graphics primitive and is larger than a pixel.

14. (Once Amended) The method of claim 12, wherein the determining step (a) further comprises the step of receiving a plurality of values for corner vertices of the polygonal portion and arithmetic edge functions [related to the graphic primitive having a coordinate reference frame located at a geometric center of the polygonal portion], each of the arithmetic edge functions corresponding to an edge of the graphics primitive.

20. (Once Amended) An electronically-readable medium having embodied thereon a program, the program being executable by a machine to perform method steps for identifying pixels inside graphics primitives of a raster image, the method steps comprising:

(a) determining whether a polygonal portion of the raster image is at least partly inside the graphics primitive by using a coordinate reference frame located at a geometric center of the polygonal portion;

(b) dividing the polygonal portion into a predetermined number of polygonal subportions if the polygonal portion is at least partly inside the graphics primitive;

(c) determining whether the polygonal subportion is at least partly inside the graphics primitive for each polygonal subportion; and

(d) dividing the polygonal subportion into a predetermined number of polygonal subportions if the polygonal subportion is at least partly inside the graphics primitive and the polygonal subportion is larger than a pixel[,].

22. (Once Amended) A method of identifying pixels inside a graphics primitive of a raster image comprising the steps of:

selecting a tile including a pixel;

defining a coordinate reference frame located at a geometric center of the tile;

determining if a portion of the tile is within the graphics primitive;

dividing the tile into subtiles if a portion of the tile is within the graphics primitive; and

recursively dividing each subtile having a portion within the graphics primitive until the subtile is equal in size to a pixel.

26. (Once Amended) An electronically-readable medium having embodied thereon a program, the program being executable by a machine to perform method steps for identifying pixels inside graphics primitives of a raster image, the method steps comprising:

selecting a tile including pixels;

defining a coordinate reference frame located at a geometric center of the tile;

determining if a portion of the tile is within the graphics primitive;

dividing the tile into subtiles if a portion of the tile is within the graphics primitive; and

recursively dividing each subtile having a portion within the graphics primitive until the subtile is equal in size to a pixel.